

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

SIMPLE UNIVERSAL HASH FOR PLAINTEXT AWARE ENCRYPTION

CROSS-REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Provisional Application
5 Serial No. 60/508,015 (Attorney Docket No. YOR920030534US1 (8728-664)), filed October 1, 2003, and entitled "SIMPLE UNIVERSAL HASH FOR PLAINTEXT AWARE ENCRYPTION", which is incorporated herein by reference in its entirety.

10 BACKGROUND OF THE INVENTION

The present invention relates generally to hashing algorithms, and in particular, to universal hashing algorithms for Plaintext aware encryption.

Cryptographic systems are known in the data processing art. In general, these systems operate by performing an encryption operation on a
15 Plaintext input message by using an encryption key and a symmetric key block cipher, and producing a Ciphertext message. The encrypted message may then be stored on an insecure device. The stored message may be decrypted with the corresponding decryption operation using the same key, to recover the Plaintext message. Since the same key is used for both the

Encryption and decryption of the message, the process is referred to as a "symmetric key" process.

Although the above encryption hides the Plaintext from an adversary, one may want to store data in an insecure and/or unreliable device and later
5 check to determine if the data was not deliberately or accidentally modified. To this end, a universal hash of the data is computed. Since the hash is a comparatively small piece of data relative to the data stored, the user will store the data and save the hash in a secure location to prevent stored data modification. When retrieving the data at a later time, the user would
10 regenerate the hash on the retrieved data, and compare it with the original hash for authenticity. Here, "universal hash" refers to the fact that the hash is key dependent, with the further property that the probability is extremely small that two messages, whether random or generated by someone who is not
15 privy to the key of the hash, will hash to the same value.

If a Ciphertext consists of several blocks, a universal hash is usually constructed by a chaining mechanism, which is inherently sequential. There are alternative methods such as a universal message authentication code ("UMAC"), which, however, require a large amount of key material.

Accordingly, what is needed is a universal hash for Plaintext-aware
20 encryption that has low-complexity and does not require a large amount of

key material.

SUMMARY OF THE INVENTION

The above and other drawbacks and deficiencies of the prior art are
5 overcome or alleviated by a simplified universal hash for Plaintext-aware encryption.

A simple universal hash apparatus and method include input means
for inputting at least one of a plurality of Plaintext blocks into an integrity
aware encryption scheme using at least one of two secret keys to obtain a
10 plurality of Ciphertext blocks; Plaintext checksum means for computing a
Plaintext checksum value from the said plurality of Plaintext blocks;
Ciphertext checksum means for processing said plurality of Ciphertext blocks
and a third key to obtain a Ciphertext checksum; and combination means for
combining the said Plaintext checksum and the said Ciphertext checksum to
15 obtain the simple universal hash value.

These and other aspects, features and advantages of the present
disclosure will become apparent from the following description of exemplary
embodiments, which is to be read in connection with the accompanying
drawings.

20

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may be better understood with reference to the following exemplary figures, in which:

Figure 1 shows a block diagram of a conventional block encryption
5 cryptographic method that operates on a Plaintext message;

Figure 2 shows a block diagram of a conventional integrity-aware encryption scheme;

Figure 3 shows a block diagram defining the Simple Universal Hash Function in accordance with a preferred embodiment of the present
10 disclosure; and

Figure 4 shows a block diagram of the Keyed Selector using key k3 in accordance with the embodiment of Figure 3.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

15 The present disclosure relates to a method and apparatus for generating a simple universal hash value of Ciphertexts produced using an integrity aware encryption scheme. Method embodiments provide for generating a cryptographic authentication code in a simple manner for Ciphertexts, which have been generated by a Plaintext aware encryption
20 scheme or encryption schemes with built in checks, or, in general, any multi

block encryption scheme where block number sensitivity is built into the Ciphertext.

Exemplary embodiments of the present disclosure are described and attained with encryption and/or decryption methods of block ciphers, including
5 embodiments realizable using a program of instructions executable by a machine to perform method steps according to the present disclosure.

An embodiment of the present disclosure defines a new class of universal hash functions computed on a sequence of Ciphertext blocks in contexts where the blocks were computed by an encryption scheme, which
10 created Ciphertext blocks by first whitening the Plaintext blocks with material generated from a first secret key and then encrypting it using a block cipher or other cryptographic primitive using the first or a second encryption key, and whitening the output of the block cipher with material generated from the first key. For future reference, such Ciphertexts will be called Plaintext aware
15 Ciphertexts. Sometimes, such schemes are also called integrity aware encryption schemes.

Another embodiment of the present disclosure defines smaller sized universal hash function values, which can be used in situations where the allowed probability of two hash functions being the same is larger. An
20 additional embodiment of the present invention provides a method for

generation of such universal hash functions, as well as an apparatus that generates such universal hash functions.

A method according to an embodiment of the present disclosure, for implementing a universal hash function on Plaintext aware Ciphertexts, is also provided. The method includes the steps of independently generating a value from each Ciphertext block and the hash key, and then computing the exclusive-or of all the values, along with a checksum computed from the Plaintext blocks, to generate the universal hash function value.

As shown in Figure 1, a conventional block-encryption cryptographic system is indicated generally by the reference numeral 100. Here, a block of Plaintext data 101 is received by a block cipher algorithm 103, such as, for example, an algorithm complying with the Digital Encryption Standard ("DES") or Advanced Encryption Standard ("AES"). The encryption algorithm 103 is used to encrypt one block of Plaintext 101 to generate one block of Ciphertext 102. The block size is fixed at 64 bits or 128 bits in DES or AES, respectively. The block cipher uses a secret key K. The secret key K is shared between the encrypting and decrypting users. To recreate the original Plaintext block, the decrypting user uses the same key and the same block cipher to decrypt the Ciphertext 102 that was used to encrypt the original Plaintext block 101.

Turning to Figure 2, a conventional Integrity Aware Encryption scheme using an Integrity Aware Parallelizable Mode ("IAPM") is indicated generally by the reference numeral 200. In IAPM, each Plaintext block P_1, P_2 to P_m is encrypted using a block cipher, such as the block ciphers 2031 through 203n, with a key k_1 , but only after first being subjected to an exclusive-or operation with S_1, S_2 to S_m respectively.

The output of the block cipher is then exclusive-or'ed with S_1, S_2 to S_m , respectively, to produce Ciphertext blocks C_1, C_2 to C_m . The integrity of the Ciphertext is assured by generating another Ciphertext block C_{m+1} . This block is generated by first taking the checksum of the Plaintexts, which, in one embodiment, is obtained by taking the exclusive-or of all the Plaintext blocks P_1, P_2 to P_m . The checksum block is then exclusive-or'ed with S_{m+1} and then encrypted with the block cipher 103, and the output of the block cipher exclusive-or'ed with S_0 to produce C_{m+1} . The sequence S_0, S_1 , to S_{m+1} is called in the art a pairwise differentially uniform sequence or xor-universal sequence. It is generated by a function block 201 from a second key K_2 , by multiplying K_2 with index i in a Galois field, or by other such operations as understood in the art.

As shown in Figure 3, a simple universal hash function according to a preferred embodiment of the present disclosure is indicated generally by the

reference numeral 300. Here, a Plaintext group of blocks 311 is passed to an integrity-aware encryption unit 310, as well as to a checksum generator 301. A Ciphertext group of blocks 312 is produced by the encryption block 310, including an mth Ciphertext block C_m . The values of the Ciphertext blocks, 5 C_1 through C_m and C_{m+1} , are each passed to a corresponding k_3 hash key of the keys 302, with the mth Ciphertext block C_m going to an mth k_3 hash key. A hashed Ciphertext group of blocks 305 is output from the hash keys, and includes hashed Ciphertext values C_1' through C_m' and C_{m+1}' . The hashed Ciphertext group of blocks 305 is passed to an Exclusive-Or block 10 303, which Exclusive-Or's the hashed Ciphertext with the checksum produced by the checksum generator 301. The output of the Exclusive-Or block is the hash value 304.

Thus, the simple universal hash function 300 is a function of the Plaintext blocks as well as the Ciphertext blocks, and the hash key k_3 . The 15 final hash value 304 is not necessarily the size of one block of the block cipher, but may be smaller, in general. As an example, if the block cipher block size is 128 bits, as in AES, and if the hash value is only supposed to be 16 bits, then the hash key k_3 will be of size 48 bits ($48 = 128/8 * \log 8$). In general, the key size K_3 will be $128/t * \log t$, where $128/t$ is the size of the 20 hash value 304. In case the hash value is only 16 bits, a checksum 301 of 16

bits is computed from the Plaintext blocks P_1, P_2 to P_m .

In one embodiment, the checksum can be computed by taking the exclusive-or of all of the Plaintext blocks, and then taking the exclusive-or of the eight 16-bit segments in the resulting 128-bit block. The Ciphertext
5 blocks C_1 to C_{m+1} produced by any Plaintext aware encryption scheme, such as, for example, the IAPM 200 of Figure 2, are then individually processed by the keyed selector 302 to obtain 16-bit values C'_1, C'_2 to C'_{m+1} respectively using the hash key k_3 as in Figure 4, to follow. The 16-bit quantities C'_1, C'_2 to C'_{m+1} are exclusive-or'ed with each other and the 16-
10 bit checksum 301 to obtain the hash value 304. In other words, the hash value 304, which is itself 16-bits long, is obtained by taking the exclusive-or sum of C'_1, C'_2 to C'_{m+1} and the checksum 301.

It shall be understood by those of ordinary skill in the pertinent art that embodiments of the present disclosure may be realized with the above-
15 described IAPM scheme, or with any scheme of encrypting several Plaintext blocks, as long as a block number sensitivity is built in to the Ciphertexts. The block number sensitivity may be built in to the Ciphertexts using a sequence such as S_1, S_2 to S_{m+1} , which is pairwise differentially uniform or pairwise independent.

20 Turning now to Figure 4, the keyed selector 302 of Figure 3, which

uses the key K_3 , is indicated generally by the reference numeral 400. Here, values of the Ciphertext block 412 are each received by a multiplexer ("MUX") 421 through 428, respectively, using a key. For example, the Ciphertext value $c1_1$ is passed to a MUX using the key $K3_1$, the Ciphertext value $c1_2$ is passed to a MUX using the key $K3_2$, the Ciphertext value $c1_3$ is passed to a MUX using the key $K3_3$, and the Ciphertext value $c1_8$ is passed to a MUX using the key $K3_4$, as indicated by the reference numeral 428, for example. The hashed Ciphertext values are output by each respective MUX to form the hashed Ciphertext block 405, comprising hashed Ciphertext bit values $C'1_1$, $C'1_2$, $C'1_3$ through $C'1_8$, respectively.

In one embodiment, the key K_3 is $128/t \cdot \log t$ bits, where $128/t$ is the size of the final hash value 304 of Figure 3. For example, when $t = 8$, the key K_3 is 48 bits. The Ciphertext block 312 of Figure 3 is divided into 16 8-bit values $C1_1$, $C1_2$, and $C1_3$ to $C1_16$. The first 3 bits of the key K_3 are used to select a single bit $C'1_1$ from $C1_1$. The 3 bits serve as an index into the byte $C1_1$. The next three bits of K_3 are used to select one bit $C'1_2$ from the next byte $C1_2$, and so on. The last three bits of K_3 , that is the least significant bits, are used to select a bit $C'1_16$ from byte $C1_16$. The concatenation 305 of the 16 bits $C'1_1$, $C'1_2$, to $C'1_16$ constitutes the 16 bit value $C'1$.

The values $C_2', C_3' \dots C_{m+1}'$ of 305 are similarly computed using the same key K_3 and the keyed selector 302. Various other keyed selectors may be used, as long as it produces a 128/t bit value 305 using the key K_3 from 128-bit Ciphertext block 102. In particular, universal hash functions known in
5 prior art maybe used as keyed selectors.

In another embodiment the last block C_{m+1} is not used in computing the final hash value 304. In other words, the exclusive-or sum 303 is performed only on the checksum 301 and the 16 bit values C_1', C_2' to C_m' .

Although illustrative embodiments have been described herein with
10 reference to the accompanying drawings, it is to be understood that the present invention is not limited to those precise embodiments, and that various changes and modifications may be effected therein by one of ordinary skill in the pertinent art without departing from the scope or spirit of the present invention. All such changes and modifications are intended to be
15 included within the scope of the present invention as set forth in the appended claims.